要求仕様書の特性に着目した個人レビュー手法の実験的評価

An Experimental Evaluation of Individual Review Methods Focusing on Software Requirements Specifications Characteristics

岡本博幸（(株) 富士通インフォソフトテクノロジ）　内山敬太（富士通（株））
鈴木彩子（ＮＴＴアドバンステクノロジ（株））　高橋一仁（日本ノーベル（株））
西山　茂（ＮＴＴアドバンステクノロジ（株），副主査）　野中　誠（東洋大学，主査）

ソフトウェア要求をレビューする際に，より効率的な手法の適用が求められる．レビュー手法には，Ad Hoc Reading (AHR)，Checklist-Based Reading (CBR)，Perspective-Based Reading (PBR) などの手法がある．これらの有効性はレビュー担当者のスキルや欠陥の種類によって異なると考えられるため，レビュー担当者のスキルの違いがレビュー手法の有効性に与える影響や，検出しやすい欠陥のレビュー手法毎の特徴を明らかにすることが求められる．本論文では，要求仕様書の特性に着目して，AHR，CBR，PBR の３つのレビュー手法に関する有効性の評価実験を行った結果を述べる．実験の結果，開発経験の浅い被験者でも PBR を適用すると有効性が発揮されること，レビュー手法によって検出される欠陥の種類に違いがあることが分かった．

When software requirements are reviewed by developers, a more efficient reading method should be employed. There are several reading methods such as Ad Hoc Reading (AHR), Checklist-Based Reading (CBR) and Perspective-Based Reading (PBR). The effectiveness of these methods varies by the factors such as the reviewer's skill and the types of defects to be detected corresponding to the characteristics of a Software Requirements Specification (SRS). Therefore, it is needed to clarify the characteristics of the methods such as the influence of reviewer's skill, the detect-ability by defect types. This paper describes the experimental evaluation result of the three reading methods; AHR, CBR and PBR. We evaluated these methods from the viewpoint of the characteristics of SRS. The result shows that PBR is more effective than the other methods even if the reviewer's experience is not enough, and that PBR is suitable for detecting completeness-related defects.

1　はじめに

ソフトウェア要求レビューは，ユーザ要求の妥当性検討や，テスト段階で検出困難な欠陥の検出ができるなど，有効な手法である．また，レビューは，一般に，テストよりも効率的に欠陥を検出できる．Jones によると，テストのみによる欠陥検出率が 30%以下であるのに対し，設計インスペクションを追加実施することにより，平均で 65%，最大で 95%の欠陥を検出できるとしている[1]．また，インスペクションを実施すると，実施しない場合に比べ 1／3 から 1／4 の工数で欠陥の修正が行われるとしている．さらに[2]では，要求定義が正規の要求定義工程以降に変更された場合，下流工程にいくほどコストが急激に増加することを示しており，開発の早期段階でのレビューやインスペクションの重要性を示唆している．このように，レビューはソフトウェアの品質や効率に大きく影響するため，より効果的なレビュー手法の適用が求められる．

レビュー手法の有効性および効率性に関する比較研究は，いくつか行われている（[3]-[7]）．これらの研究では，チェックリストに基づくレビュー (Checklist-based Reading: CBR) と，ユーザやテスト担当者などの観点別に行うレビュー (Perspective-based Reading: PBR) を比較しているものが多い．これらの研究によると，PBR の方が CBR よりも有効であるという結果が多数報告されているが，その逆の結果も指摘されている．その原因として，レビュー担当者のスキルや，レビュー手法によって検出しやすい欠陥の種類に差異があるためと考えられる．欠陥の種類とレビュー手法の有効性に関する研究では，重大欠陥と軽微欠陥を区別したものや，構文的欠陥と意味的欠陥を区別して評価した研究など[7]がある．しかしながら，欠陥の種類とレビュー手法との関連性に関する研究は，まだ十分に行われているとは言い難い状況である．

本研究の目的は，レビュー担当者のスキルおよび欠陥の種類の違いによって，各レビュー手法の有効性にどのような影響を与えるのか，その関係を明らかにすることである．本論文では，アドホックレビュー (Ad Hoc Reading: AHR)，CBR および PBR の 3 つのレビュー手法について評価実験を行い，レビュー担当者の経験と欠陥検出率との関連性について分析する．また，IEEE 830-1998[8]に示された「良いソフトウェア要求仕様の特性」（SRS 特性と呼ぶ）に着目し，これらの手法と SRS 特性との関連性について分析する．

2　レビュー手法

レビュー手法には，AHR，CBR，PBR などの手法がある．以下にそれぞれの概要を示す．

(1) Ad Hoc Reading (AHR)

AHR は，レビュー担当者に対してレビュー観点やチェック項目などの情報やレビュー指針を一切与えず，レビュー担当者の知識やこれまでの経験に基づいてレビューする手法である．レビューを実施している組織の 35%程度は，この手法を使用しているとの報告がある[9].

(2) Checklist-Based Reading (CBR)

CBR は AHR と同様，一般的に利用されている手法であり，レビュー実施組織の 50%程度がこの手法を使用している[9]．この手法では，過去の経験などに基づいてチェックリストを作成し，それを用いて一つずつチェックしていくことで，AHR に対して欠陥の検出率を上げることが期待できる．その一方で，この手法は基本的にはレビュー担当者がチェック項目すべてを考慮しながらレビューを行なうので，多くのレビュー時間を費やしたり，チェック漏れが生じるなどの欠点がある．

(3) Perspective-Based Reading (PBR)

PBR は，シナリオを作成し，そのシナリオに基づき成果物をレビューする手法である．ここでいうシナリオとはレビュー実施時のシナリオのことを示し，レビュー担当者が，利用者，設計者，テスト担当者などそれぞれの立場（観点）でレビューを実施する．1 回のレビューでは，一人当たり一つの観点でレビューを実施するため，複数人集まる会議形式レビューにおいては，レビュー観点の漏れが少なくなり，レビュー時間の短縮にも繋がる．ただし，すべての観点から何度もレビューを行うため，レビュー担当者が一人の場合には時間がかかる．

3　評価実験

3.1　レビュー対象の要求仕様書

本研究では，[10]で使用された航空機の飛行計画作成・管理システム FPM (Flight Planning and Management)のニーズ記述書とソフトウェア要求仕様書（以後，SRS）を元に編集し直したものを使用した．具体的には，仕様の簡潔化，記述項目の整理を行い，短いレビュー時間で被験者が本質的なレビューができる形式に整えた．付録Aおよび付録Bに，本研究で使用したニーズ記述書および SRS をそれぞれ示す．

3.2　欠陥の埋め込み

本実験では，ニーズ記述書には正しい情報のみが記述されていることを前提として，付録Cに示した 21 個の欠陥を SRS に埋め込んだ．これらの欠陥は全て付録Dに示される[8]の SRS 特性と対応づけられている．埋め込まれた欠陥は，非あいまい性[1] (unambiguous)，完全性 (complete)，無矛盾性 (consistent) といった一般的なもので，検証可能性(verifiable) のような比較的高度なものや，修正可能性 (modifiable) といった SRS の書式に関するものは含まれていない. 表 1 に本実験で埋め込んだ欠陥の分類を示す．21 個の欠陥のうち，約 7 割が完全性 (complete) に関する欠陥である．残りの 3 割は非あいまい性 (unambiguous)，無矛盾性 (consistent) に関する欠陥である．妥当性 (correct) に関する欠陥は 1 件と少ない．

---

[1] SRS 特性の日本語名称は，文献[11]の訳語を用いている．

埋め込まれた欠陥は，全て一意の SRS 特性に分類できるわけではなく，複数の SRS 特性の要素を含んでいる場合が少なくない．この場合，本研究ではもっとも関連が強いと思われる SRS 特性に当てはめることで，欠陥を一意に分類した．

表 1　SRS 特性別の欠陥数

妥当性（correct）	1
非あいまい性（unambiguous）	2
完全性（complete）	14
無矛盾性（consistent）	4
合計	21

3.3　被験者

AHR の被験者は，日科技連主催のソフトウェア要求開発入門セミナーの参加者であり，合計 27 名の方々に参加いただいた．4 割の被験者はソフトウェア開発者，3 割の被験者はプロジェクトマネジメント担当者であった．その他は IT アーキテクト，マーケティングなど様々であるが，いずれも数が少なかった．一方，CBR や PBR の被験者は，ある企業向けに行なったソフトウェア技術研修の受講者であり，入社 2 年以上のソフトウェア開発者が対象であった．参加人数は，CBR が 25 名，PBR が 26 名であった．

表 2 は，AHR，CBR，PBR のそれぞれの開発経験（AHR のみ要求分析の経験年数も調査）を調査した結果である．AHR の被験者は，CBR や PBR の被験者に比べて経験が豊富であった．また，図 **1**(a)は，AHR，CBR，PBR のそれぞれの開発経験の分布である．AHR の被験者は 20 年以上の経験者まで均等に分布しているのに対し, CBR は 10 年未満に集中, PBR は 5 年未満に集中していた．図 **1** (b)は，AHR の要求分析の経験の分布である．開発経験と比較すると短いが，半数以上の被験者が 5 年以上の要求分析を経験している．さらに，AHR では SWEBOK[12]のソフトウェア要求の知識度についての調査をしており，約半数の人は内容を十分には理解していなかったが，もう半数の人は概要を理解している程度であった．

表 2　被験者の経験年数

	経験年数（AHR）	要求分析年数（AHR）	経験年数（CBR）	経験年数（PBR）
平均	9.2	4.4	4.1	2.5
最大	20	15	11	4
最低	0	0	2	2
標準偏差	7.1	5.5	2.6	0.6

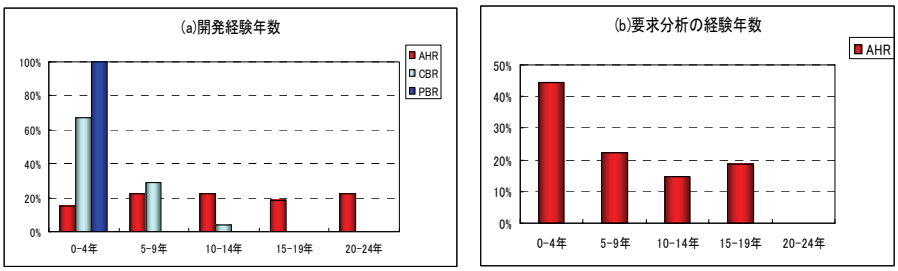

図 1　被験者の経験[(a) 開発経験, (b)要求分析の経験]

3.4　実験方法

被験者に対してニーズ記述書と欠陥が埋め込まれている SRS を配布し，要求仕様に関するレビューを行ない，その結果（指摘事項および対処方法）を回答してもらう形式をとった．AHR の被験者

には，我々からレビューの観点など特に何も言及せず，被験者各自の経験に基づいてレビューを実施してもらった．一方，CBR や PBR については，予めこちらで準備したチェックリストや観点別のチェックリストを使用し，これらにしたがってレビューを実施してもらった．

付録Eに CBR で使用した 16 個のチェック項目を含むチェックリスト, 付録Fに PBR で使用したレビュー観点別チェックリストを示す．レビューの観点は，大きく利用者／顧客の観点，設計者の観点，テスト担当者の観点の 3 つに分類し，主に付録Eで示した CBR のチェック項目がそれぞれの観点に振り分けられている．ここで，CBR と PBR のレビュー項目で大きく異なるのは，テスト担当者の観点およびその項目が PBR に追加されていることである．すなわち，測定単位，正確性，精度といった詳細をイメージさせるような項目が含まれている．

3.5　レビュー時間の違いによる欠陥検出数の調整

本実験では，セミナー受講者の協力を得て実施したものであったため，レビュー時間などの実験条件が統一できていない．とくに AHR の被験者は 40 分のレビュー時間であったのに対して，CBR および PBR では 60 分であった．また，AHR の被験者には，レビューが終了していない被験者も多く見られた．これらの違いを正規化するため，以下を考慮する．

レビュー工数と欠陥発見数との間には，テストの場合と同様にゴンペルツ曲線のような関係を示すことが推測でき, レビューの最終段階では欠陥が収束する傾向を示すものと考えられる. しかし，本研究のようにレビュー時間が短く，欠陥が飽和していない状態では，ほぼ線形的に欠陥が検出されるものと仮定できる. そこで, 本研究で欠陥検出件数や検出率を議論する場合には, CBR, PBR と AHR のレビュー時間の比である 1.5 倍を単純に AHR の結果に乗じた値を用いることにする．

4　実験結果

4.1　開発経験，要求分析の経験と欠陥指摘率

AHR，CBR，PBR の実験では，それぞれ被験者が異なるため，必然的に開発経験や要求分析の経験が異なってくる．[13]の実験では，経験年数と欠陥の有効指摘率との間には，ほとんど相関が見られず，有効なスキル評価指標にはならないとしている．その一方で，一般的な結論として議論することの危険性も指摘しており，本研究においても同様な実験を試みて検証してみることにする．

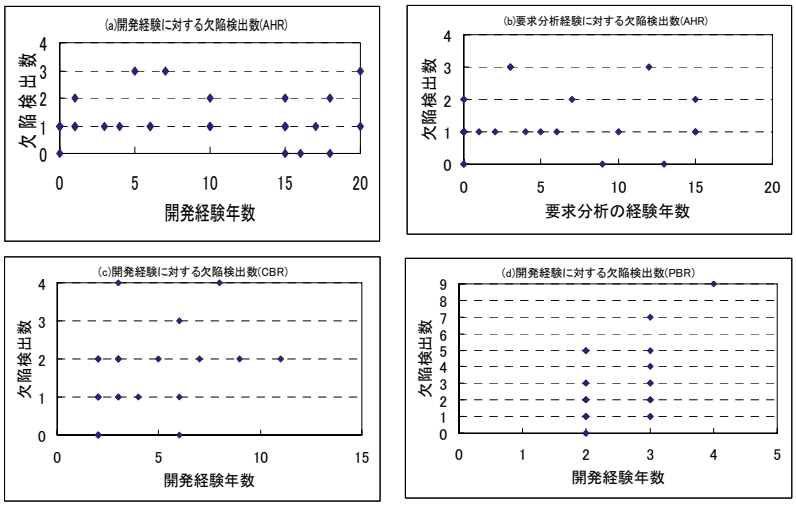

図 2　被験者の経験に対する欠陥検出数

図 2 は，開発経験，要求分析の経験と欠陥検出数との散布図である．AHR では，どちらもかなりばらつきがあり，相関係数も 0.08，0.02 とかなり低い値である．一方，CBR や PBR では，相関

係数はそれぞれ 0.40, 0.53 であり，弱いながらも経験と欠陥検出数の間に相関が見られた．これは，経験が豊富な人が必ずしも多くの欠陥を指摘できる訳ではないが，経験が浅い段階（特に 5 年未満）では，業務そのものを覚えるフェーズにあり，ある程度経験に依存する部分があることを示唆している．事実，AHR で 5 年未満のデータのみを抽出してみると，開発経験，要求分析の経験と欠陥検出数との相関係数が，それぞれ 0.35，0.25 と上昇傾向を見せている．

4.2　レビュー手法の比較

前述したとおり，AHR については，以後，欠陥検出数（率）に 1.5 を乗じた値（AHR×1.5）で議論を行なう．

図 3 に，レビュー手法毎の欠陥検出率の分布を示す．ここでの欠陥検出率とは，付録Cの埋め込み欠陥を検出できた割合であり，これ以外の指摘についてはカウントされていない．図 3 のボックスの境界線は，第一および第三四分位点，ボックス内の線はメジアンを示す．AHR と CBR ではメジアンを示す線が消えているが，これは AHR では第一四分位点とメジアン，CBR では第三四分位点とメジアンがそれぞれ一致しているためである．ボックスの上（あるいは下）の点線は，ボックスの上（あるいは下）から四分位範囲の 1.5 倍の範囲内で，データが得られた最高点（あるいは最低点）までの幅を示す．点線の外側の○印は，その範囲外にあるデータを表している[2].

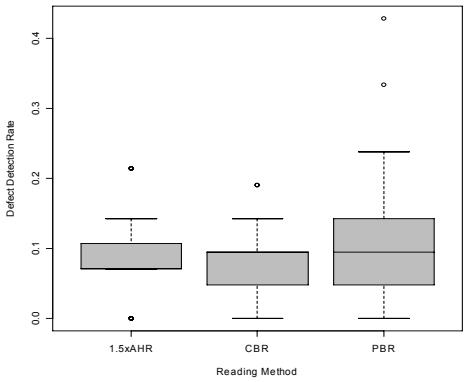

図 3　レビュー手法毎の欠陥検出率比較

レビュー時間が短かった影響で，どのレビュー手法においても欠陥検出率が低い．その関係上，レビュー手法による欠陥検出率に顕著な差は見られないが，CBR に比べ PBR の優位性が見受けられる．また PBR では，2 名の被験者が極めて高い割合で欠陥を検出していることが分かる．一方，AHR と CBR では，それぞれ第一，第三四分位点とメジアンの値が一致しており，他との優位性比較が不明確である．そこで，前提として，AHR よりも CBR, CBR よりも PBR が手法としての成熟度が高いであると考えられるので，以下の 3 つの仮説を立て，平均値の差の検定（片側検定）を行った．

＜仮説 1＞AHR よりも CBR の方が平均検出率が高い

＜仮説 2＞AHR よりも PBR の方が平均検出率が高い

＜仮説 3＞CBR よりも PBR の方が平均検出率が高い

その結果，仮説 1：P 値＝0.825，仮説 2：P 値＝0.0574，仮説 3：P 値＝0.013　となった．すなわち，有意水準 5% において，仮説 3 は採択，仮説 1 および 2 は棄却されたことになる．

仮説 1 では AHR より CBR の方が平均検出率が高くなると想定したが，本研究では反対の傾向を示した．これは，欠陥検出数（率）に単純に 1.5 を乗じたことが原因ではなく，被験者の経験に依

[2] この表現方法は一般的なものであり，分布の把握，分布の比較，外れ値の発見が容易になる．

存するものと考えられる．AHR では，被験者の 8 割強は 5 年以上の開発経験を持ち，半数以上は 5 年以上の要求分析の経験を持っている．一方，CBR の被験者は 6 割以上が開発経験 5 年未満であるので，要求分析の経験はさらに多い割合で 5 年未満であることが容易に予測できる．事実，事前アンケートの結果では，要求分析の概要を知っている程度の知識の被験者が大半で，実務で行なっている人は皆無に近いように思える．すなわち，5.1 節で述べたように，5 年未満の経験は欠陥検出率に影響を与える可能性があり，今回はこれが原因となったものと考えられる．

仮説 2 では，P 値が極めて 0.05 に近く，また，AHR グループの方が経験年数が高いことを考慮すると，同条件であれば PBR の方が平均検出率が高くなると考えられる．

4.3 SRS 特性毎の比較

図 4 に SRS 特性に基づいたレビュー手法の比較を示す．どの SRS 特性をとっても，PBR の結果が最も欠陥を検出できていることが分かる．特にここで注目したいのは，完全性 (complete) に関する検出率が 16%であり，他と比較して検出率が高いことである．PBR でのレビュー観点には，測定単位や精度に関する項目など，テスト担当者の観点が含まれており，これにより自らが実際にテストを行なう想定で，より具体的に，かつ，より詳細なレビューが実現できたものと考えられる．

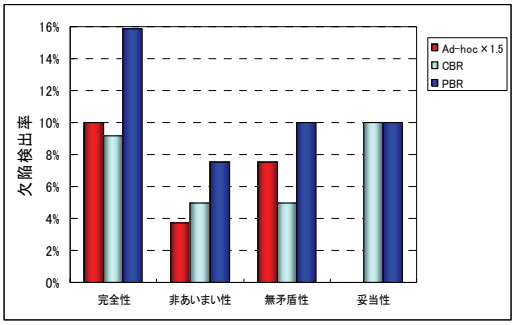

図 4　IEEE の SRS 特性に基づいたレビュー手法の比較

[14]で提唱するテスト駆動開発のように，開発の初期段階でテストケースを想定しておくことは，品質確保の観点において有効な手段である．デシジョンテーブルを用いて予めテストケースを設定しておき，その成果物をレビューする TCBR (Test Case Based Reading) と従来の CBT とを比較した研究がある[15]．この研究では，TCBR の効率性および有効性の高さが確認されており，初期段階でのテストケース設定の重要性が伺える．仮に多大な工数をかけて詳細なテストケースを設定しなくても，本研究のようなテスト担当者の観点を挙げるだけでも，充分に効果が出せたことは，一つの大きな成果と考えられる．

また，妥当性 (correct) に関する欠陥が，AHR では検出されなかったのに対して，CBR および PBR では検出されている．妥当性 (correct) に関するチェック項目は，CBR，PBR の両方に記述されており，これが検出に繋がった可能性が高い．ただし，本研究における妥当性 (correct) に関する欠陥は 1 件と非常に少なく，詳細な議論を行うには至らない．しかしながら，妥当性 (correct) に関する欠陥は下流工程にいくほど検出が困難になると考えられるため，この種の欠陥は SRS のレビュー段階で取り除かれるべきであり，その意味からも，CBR や PBR が SRS の品質確保に有効であると期待できる．

5　考察

5.1 要求分析の経験

本研究では，開発経験や要求分析の経験と欠陥検出率との相関を調査した．その結果，[13]の実験で示されているとおり，経験年数が有効なスキル評価指標にはならないことが確認できた．ただ

し，経験年数が 5 年未満の場合には弱い相関が見られ，経験に左右される可能性を示唆している．経験が浅い段階では，レビューの観点やポイントを身に付けながら業務を進めていくので，まだスキルが確立されていない発展途上の段階と言える．これゆえ，経験の差がスキルの差となって現れたものと考えられる．

5.2 レビュー手法による有効性

本研究では，被験者の経験年数を考慮すると，PBR によるレビューの有効性を示すことができた．IEEE の SRS 特性では，特に完全性 (complete) に対する欠陥の発見の効果が大きく示された．これは，PBR の要素の中には，テスト担当者の観点でのレビューが含まれており，これにより具体的，かつ，詳細なレビューが実現できたものと思われる．テスト駆動開発は，最近注目の開発方法論でもあり，PBR を導入することで，より高品質な開発が期待できる．また，PBR を導入することで，経験や知識がない人でも，高い精度で欠陥が検出できることが示されたことになる．

AHR と CBR とでは，CBR に有効性を発見することができなかった．これは，AHR の被験者のほとんどが 5 年の開発経験，要求分析の経験を持つのに対して，CBR ではほとんどが 5 年未満の経験しかないことによるものと推測できる．

5.3 課題

本研究では，PBR について，ある一定レベルの有効性が確認できた．ただし，被験者の職種や経験の違い，またレビュー時間の違いなどから，一部，仮定や推測が入り正確な議論に及んでいないところがある．本研究では，様々なセミナーや研修からデータを収集した都合があり，このような違いが生じたが，今後はこれらを極力統一する必要がある．また，SRS には IEEE の SRS 特性に基づいて欠陥を埋め込んではいるが，かなり完全性 (complete) に偏った結果になった．今後は，SRS 特性についても均一化を図る必要がある．

AHR と CBR とで CBR に有効性を発見することができなかった理由として，開発経験，要求分析の経験を挙げている．その他の要因として，チェックリストの項目の多さが挙げられる．本研究では，16 項目のチェックリストを使用しており，これらの項目全てを念頭に置きながらレビューを行うことは，困難な作業であったと考えられる．今後はチェック項目を絞った実験も必要かと思われる．

PBR では平均的に欠陥検出率が高かったもの，図 4(d)に示すように，同じ開発経験で差が開く結果になった．これは，レビュー手法によるものが大きいのか，被験者のものともとの素養が大きいのか，現段階ではその要因を言及できず，これについても今後の課題といえよう．

6 おわりに

本研究では，ソフトウェアのニーズ記述書と SRS を対象に AHR，CBR，PBR というレビュー手法を用いて，有効性の比較実験を行なった．それぞれの手法において，被験者プロフィールの違いなどがあり，全ての条件を統一することはできなかったが，PBR による欠陥検出の有効性および摘出しやすい欠陥の特性について示すことができた．

今後の課題としては，被験者プロフィールを統一させたり，チェックリストなどのツールを洗練させて再度実験を行い，本研究の結果をより確実なものとする検証が必要となる．

謝辞 実験に協力くださった被験者の方々に深く感謝の意を表します．

参考文献

[1] Jones, C.: *Estimating Software Costs*, McGraw-Hill, 1998. (富野壽監訳: ソフトウェア見積りのすべて, 構造計画研究所, 2001.)

[2] Jones, C.: *Software Quality: Analysis and Guidelines for Success,* The Coriolis Group, 2000. (富野壽監訳:, ソフトウェア品質のガイドライン, 構造計画研究所, 1999.)

[3] Thelin, T., Andersson, C., Runeson, P. and Dzamashvili-Fogelström, N.: “A Replicated Experiment of

Usage-Based and Checklist-Based Reading," *Proc. of 10th Intl. Symposium on Softw. Metrics*, IEEE Computer Society, 2004, pp.246-256.

[4] Bernárdez, B., Genero, A. and Toro, M.: "A Controlled Experiment for Evaluating a Metric-Based Reading Technique for Requirements Inspection," *Proc. of 10th Intl. Symposium on Softw. Metrics*, IEEE Computer Society, 2004, pp.257-268.

[5] Cantone, G., Colasanti, L., Abdulnabi, Z.A., Lomartire, A. and Calavaro, G.: "Evaluating Checklist-Based and Use-Case-Driven Reading Techniques as Applied to Software Analysis and Design Artifacts," *Empirical Methods and Studies in Software Engineering*, Springer-Verlag New York, 2003, pp.143-165.

[6] Thelin, T., Erlansson, M., Höst, M. and Wohlin, C.: "Experimentation with Usage-Based Reading," *Empirical Methods and Studies in Software Engineering*, Springer-Verlag New York, 2003, pp.193-207.

[7] 松川文一, Sabaliauskaite, G, 楠本真二, 井上克郎 : UML で記述された設計仕様書を対象としたレビュー手法 CBR と PBR の比較評価実験, オブジェクト指向最前線 2002, 近代科学社, 2002, pp67-74.

[8] IEEE Recommended Practice for Software Requirements Specifications, IEEE Std. 830-1998, 1998.

[9] Ciolkowski, M., Laitenberger, O. and Biffl, S.: "Software Reviews: The State of the Practice," *IEEE Software*, Nov/Dec, 2003, pp.46-51.

[10] TSPi Faculty Workshop 資料, Carnegie Mellon University, 2000.

[11] 大西淳, 剛健太郎: 要求工学, 共立出版, 2002.

[12] Abran, A. and Moore, J.W. ed.: *SWEBOK: Guide to the Software Engineering Body of Knowledge,* John Wiley & Sons, 2001. (松本吉弘監訳: ソフトウェアエンジニアリング基礎知識体系ーSWEBOKー, オーム社, 2003.)

[13] 野中誠ほか: 仕様書レビューによる個人スキル評価 ー要求分析スキルモデルー,第 19 年度ソフトウェア品質管理研究会分科会報告書, 日本科学技術連盟, 2004, pp191-209.

[14] Beck, K.: *Test-Driven Development: By Example*, Addison-Wesley, 2002. (長瀬嘉秀監訳, テスト駆動開発入門, ピアソンエデュケーション, 2003.)

[15] 野中誠: 設計・ソースコードを対象とした個人レビュー手法の比較実験, 情報処理学会研究報告 SE-146-4, 2004.

付録A

FPM システム ニーズ記述書

FPM (Flight Planning and Management) システムは、航空管制区画（区画と呼ぶ）内を飛行する飛行機の飛行計画を管理するシステムである。FPM システムのユーザは航空管制官である。

問題領域の概要

(1) ひとつの区画は、次の特徴をもった三次元領域で構成される。
   a) 垂直方向は、FL180[3]から FL600 の範囲で区切られる。なお、飛行機は境界上を飛行することができる。
   b) 水平方向は、50 海里[4]以上 150 海里以下の矩形で表される。

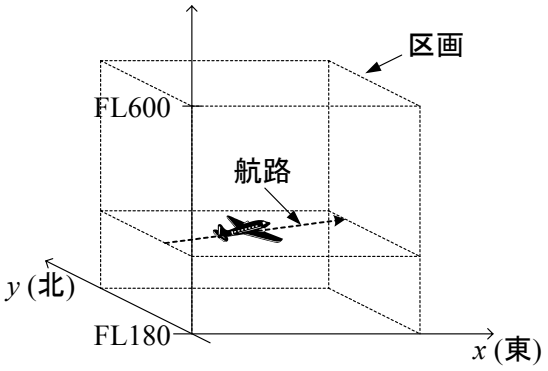

図 5　区画のイメージ

(2) 同じ高度を同時に飛行できる飛行機は、最大で 4 機までであり、互いに少なくとも 0.5 海里離れていなければならない。
(3) 異なる高度を飛行する飛行機は、少なくとも FL20 離れていなければならない。
(4) 飛行機は、区画内を飛行するための飛行計画を持つ。飛行計画は次の情報を含む。
   a) 飛行機 ID
   b) パイロットの名前
   c) 離陸時の搭載燃料（飛行時間で表記、飛行機の種類と速度に基づいて算出される）
   d) 航路。次の情報を含む。
      - 出発点座標と出発時刻
      - 高度（FL で表現、区画内を飛行している間は一定）
      - 速度（mph で表現、区画内を飛行している間は一定）
      - 針路（角度で表現、真北を 0 度とする）
      - 終点座標と終点時刻
(5) FPM システムに入力された飛行計画は、上記(2)と(3)の制約を満たしたときに承認される。
(6) FPM システムは、同一の飛行機に対して複数の飛行計画を保持できる。ただし、それらは区画内を同一時刻に飛行する飛行計画であってはならない。

---

[3] 1FL (Flight Level) = 100 フィート
[4] 1 海里 = 約 1.8km

機能ニーズ

(1) 区画の初期化

(2) 飛行計画の入力

- 入力された飛行計画をチェックし、問題が発見されなかった場合は、承認メッセージを出力し、その飛行計画を FPM システムに登録する。問題が発見された場合は、承認できない理由を記述したメッセージを出力する。
- 飛行計画の入力には、対話式による個別入力と、ファイルによる一括入力がある。

(3) 飛行計画のチェック

- 飛行計画の正確性および登録済み飛行計画に対する整合性をチェックする。

(4) 飛行計画の削除

- 指定した飛行計画を削除する。

(5) 飛行計画の表示

- 指定した飛行機の飛行計画を出力する。

(6) 飛行機の検索

- 時刻を指定すると、その時刻に区画内を飛行する飛行機のリストを出力する。
- 時刻と高度を指定すると、その時刻・高度に区画内を飛行する飛行機のリストを出力する。

(7) 高度の変更

- 指定した飛行計画の高度を変更する。

(8) 飛行機間距離の出力

- 指定した 2 機の飛行機について、異なる高度の場合は飛行機間の高度差を出力する。同じ高度の場合は、飛行機間の最小距離とその時刻を出力する。

(9) 指定距離以下となる飛行機の組合せの検索

- 距離を入力すると、高度が同じ飛行機について、飛行機間の最短距離が指定距離以下となる飛行計画を持つ飛行機の組合せと、その時刻を出力する。

非機能ニーズ

(1) 時刻は、HHMM 表記とする。HH は時刻の 24 時間表記、MM は分である。例えば、午後 3 時 3 分は 1503 と表す。

(2) 搭載燃料は自然数で表す。

(3) 高度（FL）は自然数で表す。

(4) 水平方向の距離（海里）は小数第 2 位まで表す。

(5) 速度（mph）は小数第 1 位まで表す。

(6) 針路（度）は小数第 1 位まで表す。

(7) FPM システムは、1 日分の飛行計画が管理できればよい。

(8) FPM システムは、一つの区画を維持管理できればよい。

(9) FPM システムは、応答時間に関する制約はない。

(10) 飛行計画が承認されていない飛行機が、区画内に進入することはない。

付録B

FPM システム ソフトウェア要求仕様書

1 はじめに

1.1 目的

この文書の目的は、FPM (Flight Planning and Management) システムのソフトウェア要求を定義することである。

1.2 文書構成

本文書には、システム概要 (2 章)、機能要求 (3 章)、データ定義 (4 章)、非機能要求 (5 章)、および外部インタフェース仕様 (6 章) が記述されている。

2 システム概要

FPM システムは、航空管制区画 (区画と呼ぶ) 内を飛行する飛行機の飛行計画を管理するシステムである。図 6 に区画のイメージを示す。

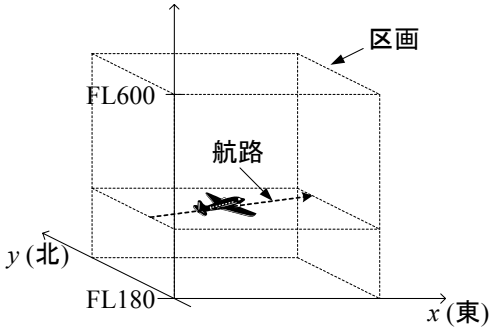

図 6 区画のイメージ

FPM システムは、一つの区画を維持管理し、区画に進入する飛行機の飛行計画を保持する。FPM システムに登録可能な飛行計画は、飛行計画に関する各種制約を満たしていなければならない。制約を満たしていない飛行計画は、FPM システムへの登録が拒否される。

FPM システムのユーザは航空管制官のみである。FPM システムは、他のシステムとは相互作用しない。FPM システムの入力インタフェースには、ユーザインタフェースと、ファイル入力インタフェースの二つがある。出力インタフェースは、ユーザインタフェースのみである。

3 機能要求

3.3 区画の初期化

FPM システムを起動した際に、ユーザから区画データの入力を受け付ける。入力された値に基づいて区画を初期化する。区画に関する制約を満たしていないデータが入力された場合は、そのデータの入力を拒否する。

3.4 飛行計画の入力

ユーザから飛行計画の入力を受け付ける。飛行計画データの正確性および整合性をチェックする (チェック機能は 3.6 節および 3.7 節)。データに問題がない場合、承認メッセージを出力し、飛行計画を登録する。問題が発見された場合、承認できない理由を記述したメッセージを出力し、飛行計画の登録を拒否する。

3.5 飛行計画リストの入力

テキストファイルに記述された飛行計画のリストを読み込み、その並び順に従って、飛行計画を逐次的に入力する。各飛行計画について、データの正確性および整合性をチェックす

る（チェック機能は 3.6 節および 3.7 節）。データに問題がない場合、承認メッセージを出力し、飛行計画を登録する。問題が発見された場合、承認できない理由を記述したメッセージを出力し、飛行計画の登録を拒否する。

なお、データファイルのフォーマットの正しさは保証されているものとし、フォーマットのチェックは行わない（データフォーマットは6.18節）。

3.6　飛行計画の正確性チェック

入力された飛行計画の正確性をチェックする。チェックは、各データ項目のフォーマットチェックを行う。また、次のチェックを行う。

- 出発点座標と終点座標は、区画の水平方向の境界面上にあるか。
- 搭載燃料は適切な値か。
- 速度は、出発点座標と終点座標の距離を、出発時刻と終点時刻の差で除算した値
- 針路は、航路に対して正しい値か。正しさの判定は図 7 に基づいて行う。

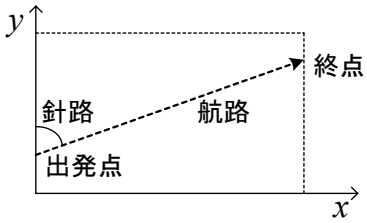

図 7　針路と航路

3.7　飛行計画の整合性チェック

入力された飛行計画の整合性をチェックする。チェック項目は次の通り。

- 区画内を同時刻に同一高度で飛行する飛行機は、最大 4 機か。
- 区画内を同時刻に同一高度で飛行する飛行機は、水平方向に少なくとも 0.5 海里離れているか。
- 区画内を同時刻に飛行する飛行機は、少なくともFL20離れているか。

なお、上記の制約に従うと、区画内には同時刻に最大で86機の飛行機が飛行できる。

3.8　飛行計画の削除

ユーザが指定した飛行機IDの飛行計画を削除する。

3.9　飛行計画の表示

ユーザが指定した飛行機IDの飛行計画を表示する。

飛行計画は、次のフォーマットで画面に出力される。

飛行機 ID： DAL111
パイロット名：　　J. Doolittle
搭載燃料： 60
出 発 点： 東10.00海里　北40.00海里
出発時刻： 1410
高　　度： FL220
速　　度： 300.0mph
針　　路： 39.8度
終　　点： 東60.00海里　北100.00海里
終点時刻： 1425

3.10　飛行機の検索（時刻指定）

ユーザが指定した時刻に区画内を飛行する飛行機について、飛行機 ID のリストを表示する。ユーザが指定した時刻と、出発時刻または終点時刻が同一の飛行計画を持つ飛行機も表示される。指定時刻が無効な値の場合、エラーメッセージを表示し、検索を行わない。

3.11　飛行機の検索（時刻・高度指定）

ユーザが指定した時刻と高度に区画内を飛行する飛行機について、飛行機 ID のリストを表示する。ユーザが指定した時刻と、出発時刻または終点時刻が同一の飛行計画を持つ飛行機も表示される。指定時刻および高度が無効な値の場合、エラーメッセージを表示し、検索を行わない。

3.12　高度の変更

「飛行計画の表示」機能（3.9 節）によりユーザが特定した飛行計画に対して、ユーザから新しい高度の数値の入力を受け付ける。高度を変更した飛行計画データの正確性および整合性をチェックする（チェック機能は 3.6 節および 3.7 節）。データに問題がない場合、承認メッセージを出力し、当該飛行計画の高度を変更する。問題が発見された場合、承認できない理由を記述したメッセージを出力し、当該飛行計画の高度変更を拒否する。

3.13　飛行機間距離の出力

ユーザが指定した 2 機の飛行機 ID について、それらの飛行機が区画内を同時刻に異なる高度で飛行する場合は、飛行機間の高度差を出力する。それらの飛行機が区画内を同時刻に同一高度で飛行する場合は、飛行機間の最小距離とその時刻を出力する。

3.14　指定距離以下となる飛行機の組合せの検索

区画内を同時刻に同一高度で飛行する飛行機のすべての組合せについて、飛行機間の距離が、ユーザが指定した距離以下となる飛行機 ID の組を出力する。

4　データ定義

4.15　区画

区画は、次の特徴をもった三次元領域で構成される。

- 垂直方向は、FL180 以上 FL600 以下の範囲で区切られる。飛行機は垂直方向の境界上を飛行することができる。自然数で表す。
- 水平方向は、50 海里以上 150 海里以下の矩形で表される。矩形上の点は（x, y）平面上の点で表され、y 方向は北を、x 方向は東を表す。小数第 2 位まで有効な固定小数点数で表す。

4.16　飛行計画

飛行計画は次の情報からなる。

- 飛行機 ID。6 桁の文字列。
- パイロットの名前。文字列。
- 離陸時の搭載燃料。自然数。
- 航路。次の情報を含む。
    - 出発点座標と出発時刻。
    - 高度。FL 単位で表す。自然数。区画内を飛行している間は一定である。
    - 速度。mph 単位で表す。小数第 1 位まで有効な固定小数点数。区画内を飛行している間は一定である。
    - 針路。角度で表す。小数第 1 位まで有効な固定小数点数。真北を 0 度とする。

- 終点座標と終点時刻。

5 非機能要求

- 時刻は、HHMM 表記とする。HH は時刻の 24 時間表記、MM は分である。例えば、午後 3 時 3 分は 1503 と表す。
- FPM システムは、1 日分（時刻 0000 から 2359 まで）の飛行計画を管理する。
- FPM システムは、応答時間に関する制約はない。
- 距離および針路に関する計算によって小数点数が得られた場合は、それぞれの数値に求められている小数点部桁数で表した値とする。
- 距離および針路に関する計算の正確性は、それぞれの数値に求められている小数点部桁数まで一致していれば正確であると判定する。
- 飛行計画が承認されていない飛行機が、区画内に進入することはない。

6 外部インタフェース仕様

6.17 ユーザインタフェース

FPM システムのユーザインタフェースは、テキストベースのメニュー型とする。初期画面は次の通り。

```
**********************************************************************
* Flight Planning and Management System v1.0
* Sat Oct 16 21:24:58 1999
**********************************************************************
区画の座標を指定します。
区画の原点（南西）のｘ座標を入力して下さい：
```

区画に関する情報がユーザから入力されたのちに、次のメニュー画面が表示される。

```
**********************************************************************
* MENU
**********************************************************************
1． 飛行計画の入力
2． 飛行計画リストの入力
3． 飛行計画の削除
4． 飛行計画の表示
5． 飛行機の検索（時刻指定）
6． 飛行機の検索（時刻・高度指定）
7． 高度の変更
8． 飛行機間距離の出力
9． 指定距離以下となる飛行機の組合せの検索
10．ヘルプ
99．終了

->
```

各メニューの仕様は省略する。

6.18　飛行計画リストファイル

ファイルは次のフォーマットに従っている。区切り記号にはカンマを使用する。

行番号	内容
1	飛行機 ID、パイロットの名前
2	離陸時の搭載燃料
3	出発点座標（*x*, *y*）、出発時刻、高度、速度、針路、終点座標（*x*, *y*）、終点時刻

飛行計画リストの例を次に示す。

行番号	内容
1	DAL111, J. Doolittle
2	60
3	10.00, 40.00, 1410, 220, 300.0, 39.85, 60.00, 100.00, 1425
4	AF237, H. Arnold
5	90
6	80.00, 20.00, 1410, 220, 400.0, 36.32, 20.00, 100.00, 1426
7	…

6.19　エラーメッセージ

省略

付録C　埋め込んだ欠陥とその種別（SRS特性）

※ web 掲載版では非公開といたします。

付録D IEEE定義のSRS特性

妥当性 (correct)	要求仕様書に記述されたすべての要求は、ソフトウェアが満たすべき要求であること
非あいまい性 (unambiguous)	要求仕様書に記述されたすべての要求は、一意にしか解釈できないこと
完全性 (complete)	機能、性能、設計上の制約、属性、外部インタフェースに関するすべての要求が記述されていること
	識別されたすべての状況における入力に対して、ソフトウェアの振る舞いが定義されていること
	要求仕様書に書かれたすべての図表にラベルがあり、用語の定義があり、測定単位が明記されていること
	TBD (To Be Determined) を含んだ要求仕様は完全ではない
	完全性に関連して、冗長な要求を含んではいけない
無矛盾性 (consistent)	他の仕様書と矛盾がなく、要求仕様書の中でも矛盾がないこと
要求の重要度と安定性のランク付け (ranked for importance and/or stability)	必須／強く要望／要望／任意
検証可能性 (verifiable)	ソフトウェア製品が要求を満たしていることを、人手または機械によりチェックできるプロセスが存在すること(ただし費用効果も考慮)
	曖昧な要求は検証不可能である
修正可能性 (modifiable)	要求に変更が生じても、容易かつ完全にSRSが修正できること
追跡可能性 (traceable)	要求の生成源が明確になっていること
	後工程で要求を参照できるようになっていること

IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications (1998).

付録E　CBRで使用したチェックリスト

チェックリストに基づく要求仕様レビュー

以下に示された項目のすべてについて、要求仕様書をレビューして下さい。

(1) 要求仕様書に記述されたすべての要求は、ソフトウェアが満たすべき要求か？
(2) すべての要求は、一意にしか解釈できないように記述されているか？
(3) データ構造、機能、性能、設計上の制約、外部インタフェースに関するすべての要求が記述されているか？
　　ただし、省略と明記されている箇所は除く。
(4) システムを利用するすべての利用者が識別されているか？
(5) 考えられるすべての状況における入力に対して、ソフトウェアの振る舞いが定義されているか？
(6) 問題領域に固有な用語は、明確に定義されているか？
(7) 測定単位や要求される正確性および精度が定義されているか？
(8) 未決事項の要求が含まれていないか？
(9) 冗長な要求記述が含まれていないか？
(10)ニーズ記述と矛盾していないか？　ただし、要求が詳細化された部分は除く。
(11)要求仕様書の内部で、他の記述と互いに矛盾する箇所はないか？
(12)要求を満たしていることを確認する手段は明らかになっているか？
(13)要求仕様が詳細に記述され過ぎていて、理解が困難になっている箇所はないか？
(14)図表は適切に利用されているか？　それらは分かりやすいか？
(15)ソフトウェア設計や実装に関する専門用語が使われていて、顧客が理解しにくい箇所はないか？
(16)ソフトウェアの内部設計に関する記述が書かれてしまっていないか？

付録F　PBRで使用したレビュー観点

観点に基づく要求仕様レビュー

以下に示されたそれぞれの観点に立って、要求仕様書をレビューして下さい。ある観点に立ってレビューしているときは、その観点にのみ焦点を当て、他の観点は考えないようにして下さい。

利用者／顧客の視点

(1)要求仕様書に記述されたすべての要求は、あなたがソフトウェア製品に期待する内容か？ ニーズ記述に基づいて判断しなさい。
(2)システムを利用するすべての利用者は識別されているか？　ニーズ記述に基づいて判断しなさい。
(3)すべての状況における入力に対して、ソフトウェアの振る舞いが定義されているか？
(4)未決事項の要求が含まれていないか？
(5)ニーズ記述と矛盾していないか？　ただし、要求が詳細化された部分は除く。
(6)問題領域に固有な用語は明確に定義されているか？
(7)ソフトウェア設計や実装に関する専門用語が使われているために、あなたが理解しにくい箇所はないか？

設計者の視点

(1)設計に必要なすべての情報（データ構造、機能、性能、設計上の制約）はすべて定義されているか？
(2)すべての外部インタフェースは明確に定義されているか？
(3)すべての要求は、一意にしか解釈できないように記述されているか？
(4)冗長な要求記述が含まれていないか？
(5)要求仕様書の内部で、他の記述と互いに矛盾しているために、設計すべき内容が理解できない部分があるか？
(6)要求が詳細に記述されすぎているために、あなたが理解できていない部分があるか？
(7)ソフトウェアの内部設計に関する記述が書かれてしまっていないか？
(8)そのアプリケーションに関する知識または概要記述に書かれた内容から考えて、要求仕様は意味をなしているか？

テスト担当者の視点

(1)測定単位や要求される正確性および精度が定義されているか？
(2)すべての機能が要求を満たしていることを確認する方法について、あなたは明確に理解できているか？
(3)すべての機能が正しい値を正しい単位で出力していることを確認する方法について、あなたは明確に理解できているか？